REMARKS

In the Office Action, the Examiner indicated that claims 1 through 22 are pending in the application and the Examiner rejected all of the claims.

Rejection under 35 U.S.C. §103

On page 4 of the Office Action, the Examiner rejected claims 1-22 under 35 U.S.C. §103(a) as unpatentable over Raj Srinivasan ("RFC 1833: Binding Protocols for ONC PRC Version 2", hereinafter "Srinivasan") in view of Simson Garfinkel et al. ("Practical UNIX & Internet Security", hereinafter "Garfinkel") and further in view of Bill Venners ("Finding Services with the Jini Lookup Service," hereinafter "Venners").

Finality of Office Action

The Examiner is respectfully requested to withdraw the finality of the Office Action. On page 2 of the Office Action, the Examiner changes his assertion that the "published name" claimed in the application is analogous, or "maps", to the "transport address" of the Srinivasan reference. Specifically, the Examiner now states that his *intention* was to show that the "published name" of the claimed invention corresponds to the "RPC Program Number" of Srinivasan (and not the transport address as stated). It is improper for the Examiner to change how he interprets the prior art vis-à-vis the claimed invention while also making the rejection final. Essentially, the Examiner issued a new grounds for rejection and made the rejection final, without a claim amendment necessitating the new grounds for rejection. This is improper.

Applicant requests, therefore, that the Examiner withdraw the finality of the present Office Action in view of this new basis for rejection.

The Present Invention

As exemplified by present independent claim 1 as currently amended, the present invention is a method of enabling a client, running on a first computing device that is connected to a second computing device, to use a service on that second computing device, comprising the steps of:

(a) a service, installed on the second computing device, registering its published name with a service broker on that second computing device; and

(b) the client sending a message to the service broker specifying the name of the service,

wherein the published name of the service conforms to a structured naming convention that both uniquely identifies the service itself and uniquely identifies the service as a service from a particular vendor, but without specifying the connection point address of that service, to enable the service broker to start up the service without the risk of a clash.

Services installed on a computing device register their published name, which conforms to a structured naming convention, such as reversed domain information, with a 'service broker' on that device. The service broker uses a single well-known port number address. When an external client, connected to the computing device that has a service broker, wants to use a service on that computing device, it sends a message to the service broker using the well known port number. The message specifies the name of the desired server and requests that the service broker inform it of the appropriate connection point (e.g. port number) to use. There is no dependency on port numbers or unstructured and arbitrary naming conventions.

<u>Applicant's Response to the Examiner's "Response to Arguments"</u>

KSR (*KSR International Co. v. Teleflex Inc.,* 127 S. Ct. 1727, 82 USPQ2d 1385 (2007) requires that an Examiner provide "some articulated reasoning with some rationale underpinning to support the legal conclusion of obviousness." Further, an Examiner must "identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does," In addition, the Examiner must make "explicit" this rationale of "the apparent reason to combine the known elements in the fashion claimed," including a detailed explanation of "the effects of demands known to the design community or present in the marketplace" and "the background knowledge possessed by a person having ordinary skill in the art."

The Examiner has not met these requirements. The 19th February 2008 response included a discussion of *Venners*. The argument made in that response was that while *Venners* discloses the concept of reverse-domain naming, this is used only to name the *type* of a Java service, and not to name the service itself. The Examiner does not address this distinction in the present action, but it is nevertheless important. The independent claims require that "the published name of the service conforms to a structured naming convention that uniquely identifies the service as a service from a particular vendor" and this is not disclosed in V*enners* (or elsewhere in the prior art).

The Examiner incorrectly summarized Applicant's argument as follows:

> *c. Venners does not teach "reverse naming" corresponding to uniquely identifying a service as being from a particular vendor.*
> (Page 3 of the OA, 3rd paragraph)

He then states that "Venners clearly discloses a "reverse form" global naming scheme that identifies object *types* as being from a particular vendor" (emphasis added) and supports this with a quote from page 8 of Venners, which states "IBM, for example, is responsible for making sure no two *types* in the *com.ibm* namespace have the same name, and *[it is]* not supposed to let the world see anything whose name doesn't start with '*com.ibm*'" (emphasis added).

What Applicant actually argued was that *Venners* does not teach using reverse naming *in the published name of the service* in order to uniquely identify the service as being from a particular vendor. Instead, *Venners* teaches using the reverse naming in the field used to describe the service's *type*. In fact, the published name of the service in *Venners* is its serviceID, nothing more than a 128-bit number that has no special connotations other than being unique to the service (see *Venners*, page 7, paragraph 3).

While *Venners*' service type field uses reverse naming to tie the service type to a particular vendor, this is not the name of the service – the name of the service is the different field serviceID. Indeed, two or more differently named services can share the same service type (see, for example, the paragraph spanning pages 8 and 9 of *Venners*). It would make no sense whatsoever to use the service type as a published name for the service, since multiple services would then share exactly the same name and the whole point of naming a service is to distinguish it from other services.

Both *Srinivasan* and *Garfinkel* disclose uniquely naming a service using an RPC program number. This number, while uniquely naming the service, does not uniquely identify the service as being provided by a particular vendor. The serviceID of *Venners* uniquely identifies the service but does not uniquely identify the vender who provided the service. Therefore, none of the cited prior art documents describes *naming* a service with a published name that "conforms to a structured naming convention *that uniquely identifies the service as a service from a particular vendor*".

These points were all made in the previous response, but appear not to have been addressed by the Examiner. Indeed, in the first paragraph of page 6 of the current Office Action, the Examiner ignores the reference to service *types* altogether, instead alleging that "Venners teaches that services are looked up by name and that service names [...] conform to a structured naming convention that uniquely identifies the service as a service from a particular vendor". This can only be correct if the term "name" is replaced throughout by "type". For the reasons given above, a service "type" is not a service "name".

Applicant refers the Examiner again to the second paragraph of page 7 of *Venners*, quoted in the 19th February 2008 response. This paragraph gives, as an example the case where "you enter a LAN environment for the first time and you want to use a printer, you don't have to figure out the printer service's registered name; instead, you just look up the services that implement a well-known printer interface. Lookup by type also ensures that Jini clients will know how to use whatever is returned from their query, because they had to have knowledge of the type before sending their query". *Venners* thus explicitly teaches away from relying upon the actual name (i.e. serviceID) of the service and instead teaches looking up services by their type.

In view of the foregoing it is respectfully submitted that *Venners* does not teach or suggest giving a service a published name that uniquely identifies its vendor and that no combination of the

prior art teaches such a feature. However, while it is believed that the claims are allowable as they stand, the independent claims have been amended to clarify that the published name both uniquely identifies the service itself *and* uniquely identifies the vendor providing the service. Note that the reverse-named service type of *Venners* is not unique to the service (for the reasons given above) and therefore cannot uniquely identify the service. The independent claims are therefore further differentiated from *Venners* by this amendment.

The Examiner has rejected claim 9 with the following explanation:

> *Regarding claim 9, Srinivasan-Garfinkel-Venners teach that* the client can request specific version of a named service *[Srnivasan: Page 13 Paragraph 6 (PMAPPROC_GETPORT)].*
>
> (Page 8 of the OA, 3rd paragraph)

Applicant does not dispute that *Srinivasan* discloses allowing the client to request a particular version of a service – this is what the “vers” parameter is for. However, the Examiner appears to have read claim 9 as providing an alternative between the client specifying the version number and the broker automatically selecting the highest version and therefore considers the claim anticipated by the presence of the former in *Srinivasan.* However, claim 9 describes a single broker that will both select a particular version if requested to, and otherwise automatically select the highest version. The wording of claim 9 (and also of corresponding claim 20) has been changed to clarify this.

Srinivasan nowhere discloses a broker that automatically selects the highest version. Instead, *Srinivasan* throughout requires the client to provide both the program number and the program version. It is essential in *Srinivasan* that the version number is provided and *Srinivasan* does not disclose the automatic selection of the highest version in the event that a version number is not provided.

Neither *Garfinkel* nor *Venners* contemplate even the possibility that multiple versions of the same service may be provided. These documents therefore do not mention the automatic selection of the highest version number available.

No combination of the cited prior art teach or suggest the features of claims 9 and 20 and it respectfully submitted that these claims are allowable.

<u>Conclusion</u>

The present invention is not taught or suggested by the prior art. Accordingly, the Examiner is respectfully requested to reconsider and withdraw the rejection of the claims. An early Notice of Allowance is earnestly solicited.

The Commissioner is hereby authorized to charge any fees associated with this communication to applicant's Deposit Account No. 50-4364.

Respectfully submitted

July 30, 2008
Date

/Mark D. Simpson/
Mark D. Simpson, Esquire
Registration No. 32,942

SAUL EWING LLP
Centre Square West
1500 Market Street, 38th Floor
Philadelphia, PA 19102-2189
Telephone: 215 972 7880
Facsimile: 215 972 4169
Email: MSimpson@saul.com